1. PURPOSE

This Program Requirements Document (PRD) identifies requirements and responsibilities for controlling the quality of computer *software* (see def.).

2. APPLICABILITY

This PRD applies to company organizations that develop, procure, modify, maintain, operate, use, or retire software. The level of analysis, documentation, and actions to demonstrate compliance with requirements shall be commensurate with the relative importance to safety, safeguards, and security; the magnitude of the hazards involved; the life-cycle stage of the facility; the programmatic mission of the software; the particular characteristics of the software complexity; economic value; and other relevant factors. Applying the specific quality assurance requirements will be based on a *graded approach* (see def.).

3. RESPONSIBILITIES

3.1 Quality Assurance

3.1.1 The quality assurance organization is responsible for providing direction and assessment of the company software quality assurance program, which includes, but is not limited to:

A. Supporting the information technologies organization in defining and disseminating company software quality assurance program requirements and assuring processes are implemented to satisfy those requirements.

B. Supporting the information technologies organization in ensuring that the company software quality assurance program complies with applicable source requirements documents.

C. Preparing, implementing, and maintaining this PRD.

D. Ensuring the contractual requirements, as specified in the appropriate contract clauses, are reflected in this PRD.

E. Providing qualified quality engineers to support the information technologies' and line organization's implementation and maintenance of the company software quality assurance program.

F. Conducting surveillances of line organizations to *verify* (see def.) implementation of the software quality assurance program.

G. Providing independent oversight of the software quality implementation to ensure compliance with applicable regulations and requirements.

3.1.2 Quality Engineer

The *Quality Engineer* (QE; see def.) is responsible for:

A. Reviewing and concurring with department or line organization software engineering documentation that implements the software quality assurance program and processes.

B. Assisting information technologies and line organizations in identifying and resolving software quality assurance *issues* (see def.).

C. Reviewing and approving software *procurement documents* (see def.).

3.2 Information Technology Organization

3.2.1 The information technologies organization through the software quality assurance group is responsible for providing central leadership and direction of the company software quality assurance program, which includes, but is not limited to:

A. Defining and disseminating company software quality assurance requirements and assuring processes are implemented to satisfy those requirements.

B. Providing interpretation of applicable source requirements for implementation in company requirements documents and procedures.

C. Implementing this PRD.

D. Ensuring the development, coordination, and maintenance of company implementing *procedures* (see def.) and establishing *processes* (see def.) and responsibilities for implementing the software quality assurance program.

E. Providing resolution and mediation support to line management for conditions that are not in compliance with software program requirements to ensure that the conditions are corrected in an appropriate and timely manner.

F. Providing *guidance* (see def.) to facilities in establishing and implementing software quality assurance program requirements.

3.3 Line Organizations

3.3.1 Line Organizations are responsible for:

A. Implementing the software quality assurance program as defined in company procedures in accordance with the requirements of this PRD.

B. Resolving software quality issues and implementing timely *corrective action* (see def.).

C. Designating individuals or organizations responsible for implementing this PRD (e.g., *technical support organization* [see def.]) and defining the interfaces with *external organizations* (see def.).

4. REQUIREMENTS

4.1 Company-wide Applications

The requirements identified in this subsection (4.1) apply to the company line organizations unless exempted by INT-17, QA PRD Introduction, Subsection 2.

4.1.1 Software Routines or Macros

4.1.1.1 *Software routines* (see def.) or *macros* (see def.) that are documented within the product in which they are developed and used and may be independently verified by visual inspection or hand calculation shall have limited requirements applied as follows:

A. Identification, including version, of the commercially available software used to develop the routine and macro.

B. Identification, including version of the software routine or macro.

C. Documentation that includes the inputs, the *computer program* (see def.) generated evidence of the programmed algorithms or equations (e.g., computer programs listings and spreadsheet cell contents), and *verification* (see def.) results showing the computer program generated correct results for a specified range of input parameters.

4.1.2 Software Procurement

4.1.2.1 PRD-5075, 4.1 Procurement Document Control, and PRD-5078, 7.1 Control of Purchased Items and Services, shall be applied to the procurement of software and *software services* (see def.).

4.1.2.2 The *purchaser* (see def.) shall be responsible for the appropriate requirements of this PRD upon acceptance of the software or related item (e.g., programmable device).

4.1.2.3 Procurement documents shall identify requirements for the Supplier's reporting of software errors to the Purchaser, and as appropriate, the Purchaser's reporting of software errors to the Supplier.

4.1.3 Software Developed Not Using This PRD

4.1.3.1 Software that has not been previously approved under a program consistent with this PRD for use in its intended application (e.g., freeware, shareware, procured commercial off-the-shelf, or otherwise acquired software) shall be evaluated in accordance with the requirements of this PRD.

4.1.3.2 The software shall be identified and controlled prior to evaluation.

4.1.3.3 The evaluation shall be performed and documented to determine adequacy to support operation and maintenance and identify the activities to be performed and the documentation that is needed.

4.1.3.4 The determination for adequacy shall be documented and shall identify as a minimum:

4.1.3.4.1 Capabilities and limitations for intended use

4.1.3.4.2 *Test plans* (see def.) and *test cases* (see def.) required to demonstrate the capabilities within the limitations

4.1.3.4.3 Instructions are provided for using the software or application within the capabilities and limitations..

4.1.3.5 Exceptions from the documentation requirements of this PRD and the justification for acceptance shall be documented.

4.1.3.6 The results of the evaluation and the performance of the actions necessary to accept the software shall be reviewed and approved.

NOTE: The resulting documentation and associated computer programs(s) shall establish the current baseline.

4.1.3.7 Revisions to previously base lined software received from organizations not required to follow the requirements in this PRD shall be evaluated in accordance with this section.

4.1.4 Software Design Process

4.1.4.1 The software *design process* (see def.) shall be documented, approved by the responsible design organization, and controlled.

4.1.4.2 The selected software engineering method shall ensure that software *life cycle* (see def.) activities are planned and performed in a traceable and orderly manner.

4.1.4.3 The appropriate software engineering elements shall define the control points and associated reviews.

4.1.4.4 Software reviews shall assure compliance with the approved software design requirements.

4.1.4.5 Reviews may be performed and documented separately or combined as appropriate to the defined software engineering method.

4.1.4.6 Reviews shall identify the participants and their specific review responsibilities.

4.1.4.7 Documentation of review comments and their disposition shall be retained until they are incorporated into the updated software.

4.1.4.8 Comments not incorporated and their disposition shall be retained until the software is approved for use.

4.1.4.9 **Identification of Software Design Requirements**

NOTE: *An item (see def.) can be called a software requirement only if its achievement can be verified and validated.*

4.1.4.9.1 Software design requirements shall address technical and software engineering requirements.

4.1.4.9.2 Software design requirements shall be identified and documented and their selection reviewed and approved.

4.1.4.9.3 Software requirements shall identify the operating system, function, interfaces, performance requirements, installation considerations, *design inputs* (see def.), and any design constraints of the computer program.

4.1.4.9.4 The software requirements shall be traceable throughout the software life cycle.

4.1.4.10 **Software Design**

4.1.4.10.1 The software design shall be documented and shall define the computational sequence necessary to meet the software requirements.

4.1.4.10.2 Software design documentation shall include (as applicable):

A. Numerical methods

B. Mathematical models

C. Control flow

D. Physical models

E. Control logic

F. Data flow

G. Process flow

H. Data structures

I. Process structures

J. Applicable relationships between data structures and process structures

NOTE: *This documentation may be combined with the documentation of the software design requirements or the computer program listings resulting from implementation of the software design.*

4.1.4.10.3 The software design shall consider the computer program's operating environment.

4.1.4.10.4 Measures to mitigate the consequences of potential problems including external and internal abnormal conditions and events that can affect the computer program shall be an integral part of the design.

4.1.4.11 **Implementation of the Software Design**

4.1.4.11.1 The software design shall be translated into computer program(s) using the programming organization's or design organization's programming standards and conventions.

4.1.4.11.2 The implementation process shall result in software products such as computer program listings and instructions for computer program use.

4.1.4.11.3 Software design verification shall be performed by competent individual(s) or group(s) other than those who developed and documented the original design, but who may be from the same organization.

4.1.4.11.4 The results of the verification activities shall be documented with the identification of the verifier indicated.

4.1.4.11.5 Software design verification methods shall include any one or a combination of *design reviews* (see def.), *alternate calculations* (see def.), and test results performed during computer program development.

4.1.4.11.5.1 Verification shall include the following items, as appropriate:

- Technical adequacy of the design approach
- Internal completeness, consistency, clarity, and correctness of the software design.
- Design traceability to the software design requirements.
- Test results.

4.1.4.11.6 The extent of verification and the methods chosen are a function of the following:

A. The complexity of the software.

B. The degree of standardization.

C. Similarity with previously proved software.

D. Importance to safety.

4.1.4.12 **Acceptance Testing**

4.1.4.12.1 Computer program *testing* (see def.) shall be performed and shall be controlled according to PRD-5082, 11.1 Test Control. Specific tests will be performed as identified in this PRD.

4.1.4.12.2 *Acceptance testing* (see def.) shall demonstrate that the computer program adequately and correctly performs all intended functions (i.e., specified software design requirements).

4.1.4.12.3 For those computer programs used in design activities, computer program test procedures shall provide for assuring that the computer program produces correct results.

4.1.4.12.4 For those computer programs used for operational control, computer program test procedures shall provide for demonstrating required performance over the range of operation of the controlled function or process.

4.1.4.12.5 Acceptance testing shall demonstrate, as appropriate, that the computer program:

A. Properly handles abnormal conditions and events as well as credible failures;

B. Does not perform adverse unintended functions; and

C. Does not degrade the system either by itself, or in combination with other functions or configuration items.

4.1.4.12.6 The computer test procedures shall provide for evaluating technical adequacy through comparison of test results from alternative methods such as hand calculations, calculations using comparable proven programs, or empirical data and information from technical literature.

4.1.4.12.7 Tests performed in support of a review can be used to complement acceptance testing. The tests and test results shall be included in the acceptance testing documentation. Such tests shall be subjected to the same criteria as the acceptance tests. These tests do not substitute for performing the comprehensive, end of development acceptance test.

4.1.4.12.8 The acceptance testing of changes to the computer program shall be subjected to selective retesting to detect unintended adverse effects introduced during the change. Such testing is intended to provide assurance that the changes have not caused unintended adverse effects in the computer program, and to verify that a modified system(s) or system components(s) still meets specified software design requirements.

4.1.4.13 **Operation**

4.1.4.13.1 After the software is approved for use and installed in the operating environment, the use of the software shall be controlled in accordance with approved procedures and instructions including the following, as applicable:

A. Application documentation (e.g., application log)

B. Access control specifications

C. Problem reporting and corrective action

D. In-use tests

E. Configuration change control process

4.1.4.14 **Maintenance**

4.1.4.14.1 Maintenance activities shall be defined to identify how enhancement requests from the user community, revisions to the software based on software design requirements, changes to the operating environment, or reported software problems that must be corrected will be controlled.

4.1.4.14.2 When applicable, in-use tests shall be performed in accordance with PRD-5082, 11.1 Test Control.

A. In-use test procedures shall be developed and documented to permit confirmation of acceptable performance of the computer program in the operating system.

B. In-use test procedures shall be performed after the computer program is installed on a different computer, or when there are significant changes in the operating system.

C. Periodic in-use manual or automatic self-check tests shall be prescribed and performed for those computer programs where computer program errors, data errors, computer hardware failures, or instrument drift can affect required performance.

4.1.4.15 **Retirement Phase**

4.1.4.15.1 During the retirement phase, the support for a software product is terminated and the routine use of the software shall be prevented.

4.1.5 Software Configuration Management

Configuration items (see def.) shall meet the requirements of this section.

4.1.5.1 Configuration items shall be maintained under *configuration management* (see def.) until the software is retired.

4.1.5.2 Software configuration management includes, but is not limited to, configuration identification, change control, and status control.

Configuration Identification

4.1.5.3 A software configuration baseline shall be established at the completion of each activity of the software design process.

4.1.5.4 Approved changes created subsequent to a baseline shall be added to the baseline.

4.1.5.5 A baseline shall define the most recently approved software configuration.

4.1.5.6 A labeling system for configuration items shall be implemented that:

A. Uniquely identifies each configuration item

B. Identifies changes to configuration items by revision

C. Provides the ability to uniquely identify each configuration of the revised software available for use

Configuration Change Control

4.1.5.7 Changes to software shall be formally controlled and documented.

4.1.5.8 The software change documentation shall include:

A. A description of the change.

B. The rationale for the change.

C. The identification of the affected software baselines.

4.1.5.9 The changes shall be formally evaluated and approved by the organization responsible for the original design, unless an alternate organization has been given the authority to approve the changes.

4.1.5.10 Only authorized changes shall be made to software baselines.

4.1.5.11 Appropriate verification activities shall be performed for the change.

4.1.5.12 The change shall be appropriately reflected in documentation and traceability of the change to the software design requirement shall be maintained.

4.1.5.13 Appropriate acceptance testing shall be performed for the change.

Configuration Status Control

4.1.5.14 The status of configuration items resulting from software design shall be maintained current.

4.1.5.15 Configuration item changes shall be controlled until they are incorporated into the approved project baseline.

4.1.5.16 The controls shall include a process for maintaining the status of changes that are proposed and approved, but not implemented. The controls shall also provide for notification of this information to impacted organizations.

4.1.6 Problem Reporting and Corrective Action

4.1.6.1 A software defect reporting and resolution system shall be implemented for software errors and failures to assure that problems are promptly reported to impacted organizations and to assure formal processing of problem resolutions.

4.1.6.2 The problem reporting and corrective action process shall address the appropriate requirements in accordance with the requirements of PRD-5087, 16.1 Corrective Action.

4.1.7 Records

4.1.7.1 Implementing procedures shall define the baseline documents that are to be maintained as records in accordance with PRD-5088, 17.1 Quality Assurance Records. These documents can be provided as separate or as combined documents.

5. DEFINITIONS

Refer to LST-199, Definitions, in the QA PRD Manual for the definitions of the following terms:

acceptance testing

alternate calculations

computer program

configuration item

configuration management

corrective action

design input

design process

design review

external organization

guidance

issues

item

macro

procedure

process

procurement document

purchaser

quality engineer

quality assurance records

service

software

software life cycle

software routine

structures, systems, and components

technical support organization

testing

test case

test plan

verification

verify

6. REFERENCES

ASME NQA-1-2000, Quality Assurance Requirements for Nuclear Facility Applications

DOE/RW-0333P, Office of Civilian Radioactive Waste Management, Quality Assurance Requirements and Description, Revision 10

7. APPENDICES

None.